The Power of Synchronisation: Formal Analysis of Power Consumption in Networks of Pulse-Coupled Oscillators*

Paul Gainer[0000−0002−5323−8501], Sven Linker[0000−0003−2913−7943], Clare Dixon, Ullrich Hustadt[0000−0002−0455−0267], and Michael Fisher[0000−0002−0875−3862]

University of Liverpool, UK
{p.gainer, s.linker, cldixon, u.hustadt, mfisher}@liverpool.ac.uk

Abstract. Nature-inspired synchronisation protocols have been widely adopted to achieve consensus within wireless sensor networks. We analyse the power consumption of such protocols, particularly the energy required to synchronise all nodes across a network. We use the model of bio-inspired, pulse-coupled oscillators to achieve network-wide synchronisation and provide an extended formal model of just such a protocol, enhanced with structures for recording energy usage. Exhaustive analysis is then carried out through formal verification, utilising the PRISM model-checker to calculate the resources consumed on each possible system execution. This allows us to investigate a range of parameter instantiations and the trade-offs between power consumption and time to synchronise. This provides a principled basis for the formal analysis of a broader range of large-scale network protocols.

Keywords: Probabilistic Verification · Synchronisation · Wireless Sensor Nets

1 Introduction

Minimising power consumption is a critical design consideration for wireless sensor networks (WSNs) [20, 1]. Once deployed a WSN is generally expected to function independently for long periods of time. In particular, regular battery replacement can be costly and impractical for remote sensing applications. Hence, it is important to reduce the power consumption of the individual nodes by choosing low-power hardware and/or energy efficient protocols. However, to make informed choices, it is also necessary to have good estimations of the power consumption for individual nodes. While the general power consumption of the

* This work was supported by the Sir Joseph Rotblat Alumni Scholarship at Liverpool, the EPSRC Research Programme EP/N007565/1 *Science of Sensor Systems Software* and the EPSRC Research Grant EP/L024845/1 *Verifiable Autonomy*. The authors would like to thank the Networks Sciences and Technology Initiative (NeST) of the University of Liverpool for the use of their computing facilities and David Shield for the corresponding technical support.

hardware can be extracted from data sheets, estimating the overall power consumption of different protocols is more demanding.

Soua and Minet provided a general taxonomy for the analysis of wireless network protocols with respect to energy efficiency [21] by identifying the contributing factors of energy wastage, for instance packet collisions and unnecessary idling. These detrimental effects can be overcome by allocating time slots for node communication. That is, nodes in a network need to synchronise their clocks and use time slots for communication to avoid packet collisions [23, 19].

Biologically inspired synchronisation protocols are well-suited for WSNs since centralised control is not required to achieve synchrony. The protocols build on the underlying mathematical model of pulse-coupled oscillators (PCOs) [15, 17, 18]; integrate-and-fire oscillators with pulsatile coupling, such that when an oscillator fires it induces some phase-shift response determined by a *phase response function*. Mutual interactions can lead to all oscillators firing synchronously.

In previous work [9] we used a *population model* [5, 6, 8] to encode information about groups of oscillators sharing the same configuration. Furthermore, we introduced *broadcast failures* where an oscillator may fail to broadcast its message. Since WSNs operate in stochastic environments under uncertainty we encoded these failures within a *probabilistic* model. Here, we extend our model with reward structures to associate different current draws with its states, thus enabling us to measure the energy consumption of the overall network. We employ the probabilistic model checker Prism [14] to analyse the average and worst-case energy consumption for both the synchronisation of arbitrarily configured networks, and restabilisation of a network, where a subset of oscillators desynchronised. We derive a metric from the complex order parameter of Kuramoto [13]. Since exact time synchronisation in real-world scenarios is not possible, it is sufficient for all oscillators to fire within some defined time window [3].

The structure of the paper is as follows. In Sect. 2 we discuss related work, and in Sect. 3 we introduce the general PCO model, from which we derive population models. Section 4 introduces the derived synchronisation metric. The construction of the formal model used for the analysis is presented in Sect. 5. Subsequently, in Sect. 6 we evaluate the results for certain parameter instantiations and discuss their trade-offs with respect to power consumption and time to synchronise. Section 7 concludes the paper.

2 Related Work

Formal methods, in particular model checking, have been successfully used to model and analyse protocols for wireless sensor systems. Heidarian et al. used model checking to analyse clock synchronisation for medium access protocols [11]. They considered both fully-connected networks and line topologies with up to four nodes. Model checking of biologically inspired coupled oscillators has also been investigated by Bartocci et al. [2]. They present a subclass of timed automata suitable to model biological oscillators, and a model checking algorithm. However, their analysis was restricted to a network of three oscillators.

We introduced a formal population model for a network of PCOs [9], and investigated both the probability and expected time for an arbitrarily configured population of oscillators to synchronise. For very small devices with limited resources, it is important to minimise the cost of low-level functionalities, such as synchronisation. Even a floating point number may need too much memory, compared to an implementation with, for example, a four-bit vector. Hence, in our model the oscillators synchronise over a finite set of *discrete* clock values.

The oscillation cycle includes a *refractory period* at the start of the oscillation cycle where an oscillator cannot be perturbed by other firing oscillators. This corresponds to a period of time where a WSN node enters a low-power idling mode. In this work we extend this approach by introducing a metric for global power consumption and discuss refinements of the model that allows us to formally reason about much larger populations of oscillators.

Wang et al. proposed an energy-efficient strategy for the synchronisation of PCOs [22]. In contrast to our work, they consider real-valued clocks and *delay-advance phase response* functions, where both positive and negative phase shifts can occur. A result of their choice of phase response function is that synchronisation time is independent of the length of the refractory period, in contrast to our model. Furthermore, they assume that the initial phase difference between oscillators has an upper bound. They achieve synchrony for refractory periods larger than half the cycle, while our models do not always synchronise in these cases, as we do not impose a bound on the phase difference of the oscillators. We consider all possible differences in phase since we examine the energy consumption for the resynchronisation of a subset of oscillators.

Konishi and Kokame conducted an analysis of PCOs where a perceived pulse immediately resets oscillators to the start of their cycle [12]. Their goal was to maximise refractory period length, while still achieving synchronisation within some number of clock cycles. Similarly to our work, they restricted their analysis to a fully coupled network. They assumed that the protocol was implemented as part of the physical layer of the network stack by using capacitors to generate pulses, therefore their clocks were continuous and had different frequencies. We assume that the synchronisation protocol resides on a higher layer, where the clock values are discretised and oscillate with the same frequency.

3 Oscillator Model

We consider a fully-coupled network of PCOs with identical dynamics over discrete time, since homogeneous wireless sensor networks are prevalent. The *phase* of an oscillator i at time t is denoted by $\phi_i(t)$. The phase of an oscillator progresses through a sequence of discrete integer values bounded by some $T \geq 1$. The phase progression over time of a single uncoupled oscillator is determined by the successor function, where the phase increases over time until it equals T, at which point the oscillator will fire in the next moment in time and the phase will reset to one. The phase progression of an uncoupled oscillator is therefore cyclic with period T, and we refer to one cycle as an *oscillation cycle*.

When an oscillator fires, it may happen that its firing is not perceived by any of the other oscillators coupled to it. We call this a *broadcast failure* and denote its probability by $\mu \in [0,1]$. Note that μ is a global parameter, hence the chance of broadcast failure is identical for all oscillators. The occurrences of broadcast failures are statistically independent, since, the parameter μ represents detrimental effects on the communication medium itself, for example fog impairing vision, or static electricity interfering with radio messages.

When an oscillator fires, and a broadcast failure does not occur, it perturbs the phase of all oscillators to which it is coupled; we use $\alpha_i(t)$ to denote the number of all other oscillators that are coupled to i and will fire at time t. The *phase response function* is a positive increasing function $\Delta : \{1,\ldots,T\} \times \mathbb{N} \times \mathbb{R}^+ \to \mathbb{N}$ that maps the phase of an oscillator i, the number of other oscillators perceived to be firing by i, and a real value defining the strength of the coupling between oscillators, to an integer value corresponding to the perturbation to phase induced by the firing of oscillators where broadcast failures did not occur.

We can introduce a refractory period into the oscillation cycle of each oscillator. A refractory period is an interval of discrete values $[1,R] \subseteq [1,T]$ where $1 \leq R \leq T$ is the size of the refractory period, such that if $\phi_i(t)$ is inside the interval, for some oscillator i at time t, then i cannot be perturbed by other oscillators to which it is coupled. If $R = 0$ then we set $[1,R] = \emptyset$, and there is no refractory period at all. The *refractory function* $\mathrm{ref} : \{1,\ldots,T\} \times \mathbb{N} \to \mathbb{N}$ is defined as $\mathrm{ref}(\Phi,\delta) = \Phi$ if $\Phi \in [1,R]$, or $\mathrm{ref}(\Phi,\delta) = \Phi + \delta$ otherwise, and takes as parameters δ, the degree of perturbance to the phase of an oscillator, and Φ, the phase, and increases the phase by δ if Φ is outside of the refractory period.

The phase evolution of an oscillator i over time is then defined as follows, where the *update function* and *firing predicate*, respectively denote the updated phase of oscillator i at time t in the next moment in time, and the firing of oscillator i at time t,

$$\mathit{update}_i(t) = 1 + \mathrm{ref}(\phi_i(t), \Delta(\phi_i(t), \alpha_i(t), \epsilon)), \qquad \mathit{fire}_i(t) = \mathit{update}_i(t) > T,$$
$$\phi_i(t+1) = \begin{cases} 1 & \text{if } \mathit{fire}_i(t) \\ \mathit{update}_i(t) & \text{otherwise.} \end{cases}$$

3.1 Population Model

Let Δ be a phase response function for a network of N identical oscillators, where each oscillator is coupled to all other oscillators, and where the coupling strength is given by ϵ. Each oscillator has a phase in $1,\ldots,T$, and a refractory period defined by R. The probability of broadcast failure in the network is $\mu \in [0,1]$. We define a *population model* of the network as $\mathcal{S} = (\Delta, N, T, R, \epsilon, \mu)$. Oscillators in our model have identical dynamics, and two oscillators are indistinguishable if they share the same phase. We therefore encode the global state of the model as a tuple $\langle k_1,\ldots,k_T\rangle$ where each k_Φ is the number of oscillators with phase Φ.

A *global state* of $\mathcal{S}$ is a T-tuple $\sigma \in \{0,\ldots,N\}^T$, where $\sigma = \langle k_1,\ldots,k_T\rangle$ and $\sum_{\Phi=1}^{T} k_\Phi = N$. We denote by $\Gamma(\mathcal{S})$ the set of all global states of $\mathcal{S}$, and

will simply use Γ when $\mathcal{S}$ is clear from the context. Fig. 1 shows four global states of a population model of $N = 8$ oscillators with $T = 10$ discrete values for their phase and a refractory period of length $R = 2$. For example $\sigma_0 = \langle 2, 1, 0, 0, 5, 0, 0, 0, 0, 0\rangle$ is the global state where two oscillators have a phase of one, one oscillator has a phase of two, and five oscillators have a phase of five. The starred node indicates the number of oscillators with phase ten that will fire in the next moment in time, while the shaded nodes indicate oscillators with phases that lie within the refractory period (one and two). If no oscillators have some phase Φ then we omit the 0 in the corresponding node.

We distinguish between states where one or more oscillators are about to fire, and states where no oscillators will fire at all. We refer to these states as *firing states* and *non-firing* states respectively. Given a population model $\mathcal{S}$, a global state $\langle k_1, \ldots, k_T\rangle \in \Gamma$ is a *firing state* if, and only if, $k_T > 0$. We respectively denote the sets of firing and non-firing states of $\mathcal{S}$ by $\Gamma^{\mathsf{F}}(\mathcal{S})$ and $\Gamma^{\mathsf{NF}}(\mathcal{S})$.

3.2 Successor States

We now define how the global state of a population model evolves over time. Since our population model encodes uncertainty in the form of broadcast failures, firing states may have more than one possible successor state. We denote the transition from a firing state σ to a possible successor state σ' by $\sigma \to \sigma'$. With every firing state $\sigma \in \Gamma^{\mathsf{F}}$ we associate a non-empty set of *failure vectors*, where each failure vector is a tuple of broadcast failures that could occur in σ. A *failure vector* is a T-tuple where the Φ^{th} element denotes the number of broadcast failures that occur for all oscillators with phase Φ. If the Φ^{th} element is $\star$ then no oscillators with a phase of Φ fired. We denote the set of all possible failure vectors by $\mathcal{F}$. Oscillators with phase less than T may fire due to being perturbed by the firing of oscillators with a phase of T. This is discussed in detail later in this section. We refer the reader to [9] for a detailed description of failure vector calculations.

A non-firing state will always have exactly one successor state, as there is no oscillator that is about to fire. Therefore, the phase of every oscillator is simply updated by one in the next time step, until one or more oscillators fire and perturb the phase of other oscillators. Given a sequence of global states $\sigma_0, \sigma_1, \ldots, \sigma_{n-1}, \sigma_n$ where $\sigma_0, \ldots \sigma_{n-1} \in \Gamma^{\mathsf{NF}}$ and $\sigma_n \in \Gamma^{\mathsf{F}}$, we omit transitions between σ_i and σ_{i+1} for $0 \leq i < n$, and instead introduce a direct transition from the first non-firing state σ_0 to the next firing state σ_n in the sequence. This is a refinement of the model presented in [9]. While the state space remains the same the number of transitions in the model is substantially decreased. Hence the time and resources required to check desirable properties are reduced. We denote the single successor σ' of a non-firing state σ by $\vec{\text{succ}}(\sigma)$. For example, in Fig. 1 we have $\vec{\text{succ}}(\sigma_0) = \sigma_1$.

For real deployments of protocols for synchronisation the effect of one or more oscillators firing may cause other oscillators to which they are coupled to fire in turn. This may then cause further oscillators to fire, and so forth, and we refer to this event as a *chain reaction*. When a chain reaction occurs it can lead to multiple groups of oscillators being triggered to fire and being *absorbed* by the initial

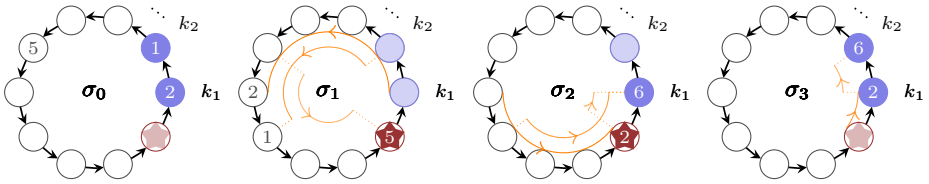


Fig. 1. Evolution of the global state over four discrete time steps.

group of firing oscillators. These chain reactions are usually near-instantaneous events. Since we model the oscillation cycle as a progression through a number of discrete states, we choose to encode chain reactions by updating the phases of all perturbed oscillators in a single time step. Since we only consider fully-connected topologies, any oscillators sharing the same phase will always perceive the same number of other oscillators firing.

For the global state σ_1 of Fig. 1 we can see that five oscillators will fire in the next moment in time. In the successor state σ_2, the single oscillator with a phase of seven in σ_1 perceives the firing of the five oscillators. The induced perturbation causes the single oscillator to also fire and therefore be absorbed by the group of five. The remaining two oscillators with a phase of six in σ_1 perceive six oscillators to be firing, but the induced perturbation is insufficient to cause them to also fire, and they instead update their phases to ten.

With every firing state we have by definition that at least one oscillator is about to fire in the next time step. Since the firing of this oscillator may, or may not, result in a broadcast failure we can see that at least two failure vectors will be associated with any firing state, and that additional failure vectors will be associated with firing states where more than one oscillator is about to fire. Given a firing state σ and a failure vector F associated with that state, we can compute the successor of σ. For each phase $\Phi \in \{1,\ldots,T\}$ we calculate the number of oscillators with a phase greater than Φ perceived to be firing by oscillators with phase Φ. We simultaneously calculate $\mathit{update}^{\Phi}(\sigma, F)$, the updated phase of oscillators with phase Φ, and $\mathit{fire}^{\Phi}(\sigma, F)$, the predicate indicating whether or not oscillators with phase Φ fired. Details of these constructions are given in [9].

We can then define the function that maps phase values to their updated values in the next moment in time. Since we do not distinguish between oscillators with the same phase we only calculate a single updated value for their phase. The *phase transition function* $\tau : \Gamma^{\mathsf{F}} \times \{1,\ldots,T\} \times \mathcal{F} \to \mathbb{N}$ maps a firing state σ, a phase Φ, and a failure vector F for σ, to the updated phase in the next moment in time, with respect to the broadcast failures defined in F, and is defined as $\tau(\sigma, \Phi, F) = 1$ if $\mathit{fire}^{\Phi}(\sigma, F)$, and $\tau(\sigma, \Phi, F) = \mathit{update}^{\Phi}(\sigma, F)$ otherwise.

Let $\mathcal{U}_{\Phi}(\sigma, F)$ be the set of phase values Ψ where all oscillators with phase Ψ in σ will have the updated phase Φ in the next time step, with respect to the broadcast failures defined in F. Formally, $\mathcal{U}_{\Phi}(\sigma, F) = \{\Psi \mid \Phi \in \{1,\ldots,T\} \wedge \tau(\sigma, \Psi, F) = \Phi\}$. We can now calculate the successor state of a firing state σ and

define how the model evolves over time. Observe that the population model does not encode oscillators leaving or joining the network, therefore the population N remains constant. The *firing successor function* $\vec{\text{succ}} : \Gamma^{\mathsf{F}} \times \mathcal{F} \to \Gamma$ maps a firing state σ and a failure vector F to a global state σ', and is defined as $\vec{\text{succ}}(\langle k_1, \ldots, k_T \rangle, F) = \langle k'_1, \ldots, k'_T \rangle$, where $k'_\Phi = \sum_{\Psi \in \mathcal{U}_\Phi(\sigma, F)} k_\Psi$ for $1 \leq \Phi \leq T$.

3.3 Transition Probabilities

We now define the probabilities that will label the transitions in our model. Given a global state $\sigma \in \Gamma$, if σ is a non-firing state then it has exactly one successor state. If σ is a firing state then to construct the set of possible successor states we must first construct $\mathcal{F}_\sigma$, the set of all possible failure vectors for σ. Given a global state $\sigma \in \Gamma$ we define $next(\sigma)$, the set of all successor states of σ, as

$$next(\sigma) = \begin{cases} \{\vec{\text{succ}}(\sigma, F) \mid F \in \mathcal{F}_\sigma\} & \text{if } \sigma \in \Gamma^{\mathsf{F}} \\ \{\vec{\vec{\text{succ}}}(\sigma)\} & \text{if } \sigma \in \Gamma^{\neg\mathsf{F}}. \end{cases}$$

For every non-firing state $\sigma \in \Gamma^{\neg\mathsf{F}}$ we have $|next(\sigma)| = 1$, since there is always exactly one successor state $\vec{\vec{\text{succ}}}(\sigma)$, and we label the transition from σ to $\vec{\vec{\text{succ}}}(\sigma)$ with probability one. We now consider each firing state $\sigma = \langle k_1, \ldots, k_n \rangle \in \Gamma^{\mathsf{F}}$, and for every successor $\vec{\text{succ}}(\sigma, F) \in next(\sigma)$, we calculate the probability that will label $\sigma \to \vec{\text{succ}}(\sigma, F)$. Recalling that μ is the probability of a broadcast failure occurring, let $\mathsf{PMF} : \{1, \ldots, N\}^2 \to [0, 1]$ be a probability mass function where $\mathsf{PMF}(k, f) = \mu^f (1-\mu)^{k-f} \binom{k}{f}$ is the probability that f broadcast failures occur given that k oscillators fire. Then let $\mathsf{PFV} : \Gamma^{\mathsf{F}} \times \mathcal{F} \to [0, 1]$ be the function mapping a firing state $\sigma = \langle k_1, \ldots, k_T \rangle$ and a failure vector $F = \langle f_1, \ldots, f_T \rangle \in \mathcal{F}$ to the probability of the failures in F occurring in σ, given by

$$\mathsf{PFV}(\sigma, F) = \prod_{\Phi=1}^{T} \begin{cases} \mathsf{PMF}(k_\Phi, f_\Phi) & \text{if } f_\Phi \neq \star \\ 1 & \text{otherwise.} \end{cases}$$

We can now describe the evolution of the global state over time. A *run* of a population model $\mathcal{S}$ is an infinite sequence $\sigma_0, \sigma_1, \sigma_2, \cdots$, where σ_0 is called the *initial state*, and $\sigma_{i+1} \in next(\sigma_i)$ for all $i \geq 0$.

4 Synchronisation and Metrics

Given a population model $\mathcal{S} = (\Delta, N, T, R, \epsilon, \mu)$, and a global state $\sigma \in \Gamma$, we say that σ is *synchronised* if all oscillators in σ share the same phase. We say that a run of the model $\sigma_0, \sigma_1, \sigma_2, \cdots$ *synchronises* if there exists an $i > 0$ such that σ_i is synchronised. Note that if a state σ_i is synchronised then any successor state σ_{i+1} of σ_i will also be synchronised.

We can extend this binary notion of synchrony by introducing a *phase coherence* metric for the level of synchrony of a global state. Our metric is derived from

the *order parameter* introduced by Kuramoto [13] as a measure of synchrony for a population of coupled oscillators. If we consider the phases of the oscillators as positions on the unit circle in the complex plane we can represent the positions as complex numbers with magnitude 1. The function $p^{\mathbb{C}} : \{1, \ldots, T\} \to \mathbb{C}$ maps a phase value to its corresponding position on the unit circle in the complex plane, and is defined as $p^{\mathbb{C}}(\Phi) = e^{i\theta_\Phi}$, where $\theta_\Phi = \frac{2\pi}{T}(\Phi - 1)$.

A measure of synchrony r can then be obtained by calculating the magnitude of the complex number corresponding to the mean of the phase positions. A global state has a maximal value of $r = 1$ when all oscillators are synchronised and share the same phase Φ, mapped to the position defined by $p^{\mathbb{C}}(\Phi)$. It then follows that the mean position is also $p^{\mathbb{C}}(\Phi)$ and $|p^{\mathbb{C}}(\Phi)| = 1$. A global state has a minimal value of $r = 0$ when all of the positions mapped to the phases of the oscillators are uniformly distributed around the unit circle, or arranged such that their positions achieve mutual counterpoise. The *phase coherence function* $\mathsf{PCF} : \Gamma \to [0, 1]$ maps a global state to a real value in the interval $[0, 1]$, and is given by

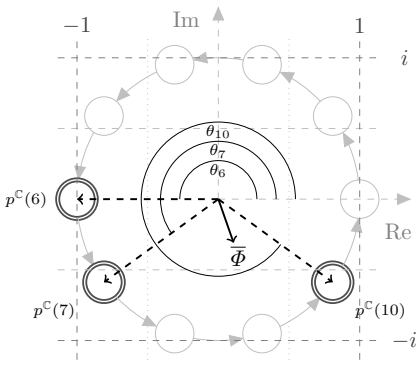


Fig. 2. Argand diagram of the phase positions for global state $\sigma_1 = \langle 0, 0, 0, 0, 0, 2, 1, 0, 0, 5\rangle$.

$$\mathsf{PCF}(\langle k_1, \ldots, k_T\rangle) = \left| \tfrac{1}{N} \textstyle\sum_{\Phi=1}^{T} k_\Phi p^{\mathbb{C}}(\Phi) \right|.$$

Note that for any synchronised global state σ we have that $\mathsf{PCF}(\sigma) = 1$, since all oscillators in σ share the same phase.

Figure 2 shows a plot on the complex plane of the positions of the phases for $N = 8$, $T = 10$, and the global state $\sigma_1 = \langle 0, 0, 0, 0, 0, 2, 1, 0, 0, 5\rangle$. The phase positions are given by $p^{\mathbb{C}}(6) = e^{i\pi}$ for 2 oscillators with phase 6, $p^{\mathbb{C}}(7) = e^{\frac{6i\pi}{5}}$ for 1 oscillator with phase 7, and $p^{\mathbb{C}}(10) = e^{\frac{9i\pi}{5}}$ for 5 oscillators with phase 10. We can then determine the phase coherence as $\mathsf{PCF}(\sigma) = |\frac{1}{8}(2e^{i\pi} + e^{\frac{6i\pi}{5}} + 5e^{\frac{9i\pi}{5}})| = 0.4671$. The mean phase position is indicated on the diagram by $\overline{\Phi}$.

5 Model Construction

We use PRISM [14] to formally verify properties of our model. Given a probabilistic model of a system, PRISM can be used to reason about temporal and probabilistic properties of the input model, by checking requirements expressed in a suitable formalism against all possible runs of the model. We define our input models as *discrete time Markov chains* (DTMCs). A DTMC is a tuple $(Q, \sigma_I, \mathsf{P})$ where Q is a set of states, $\sigma_I \in Q$ is the initial state, and $\mathsf{P} : Q \times Q \to [0, 1]$

is the function mapping pairs of states (q, q') to the probability with which a transition from q to q' occurs, where $\sum_{q' \in Q} \mathsf{P}(q, q') = 1$ for all $q \in Q$.

Given a population model $\mathcal{S} = (\Delta, N, T, R, \epsilon, \mu)$ we construct a DTMC $D(\mathcal{S}) = (Q, \sigma_I, \mathsf{P})$. We define the set of states Q to be $\Gamma(\mathcal{S}) \cup \{\sigma_I\}$, where σ_I is the initial state of the DTMC. In the initial state all oscillators are *unconfigured*. That is, oscillators have not yet been assigned a value for their phase. For each $\sigma = \langle k_1, \ldots, k_T \rangle \in Q \setminus \{\sigma_I\}$ we define

$$\mathsf{P}(\sigma_I, \sigma) = \frac{1}{T^N} \binom{N}{k_1, \ldots, k_T}$$

to be the probability of moving from σ_I to a state where k_i arbitrary oscillators are configured with the phase value i for $1 \leq i \leq T$. The multinomial coefficient defines the number of possible assignments of phases to distinct oscillators that result in the global state σ. The fractional coefficient normalises the multinomial coefficient with respect to the total number of possible assignments of phases to all oscillators. In general, given an arbitrary set of initial configurations (global states) for the oscillators, the total number of possible phase assignments is T^N.

We assign probabilities to the transitions as follows: for every $\sigma \in Q \setminus \{\sigma_I\}$ we consider each $\sigma' \in Q \setminus \{\sigma_I\}$ where $\sigma' = \vec{\mathrm{succ}}(\sigma, F)$ for some $F \in \mathcal{F}_\sigma$, and set $\mathsf{P}(\sigma, \sigma') = \mathsf{PFV}(\sigma, F)$. For all other $\sigma \in Q \setminus \{\sigma_I\}$ and $\sigma' \in Q$, where $\sigma \neq \sigma'$ and $\sigma' \notin \mathit{next}(\sigma)$, we set $\mathsf{P}(\sigma, \sigma') = 0$.

To facilitate the analysis of parameterwise-different population models we provide a Python script that allows the user to define ranges for N, T, R, ϵ, and μ. The script then automatically generates a model for each set of parameter values, checks given properties in the model using PRISM, and writes user specified output to a file which can be used by statistical analysis tools.[1]

5.1 Reward Structures

We can annotate DTMCs with information about rewards (or costs) by using a *reward structure*. A reward structure is a pair of functions, $\rho_Q : Q \to \mathbb{R}$ and $\rho_{\mathsf{P}} : Q \times Q \to \mathbb{R}$, that respectively map states and transitions to real values. By calculating the expected value of these rewards we can reason about quantitative properties of the models. For a network of WSN nodes we are interested in the time taken to achieve a synchronised state and the power consumption of the network. Given a population model $\mathcal{S} = (\Delta, N, T, R, \epsilon, \mu)$, and its corresponding DTMC $D(\mathcal{S}) = (Q, \sigma_I, \mathsf{P})$, we define the following reward structures:

Synchronisation Time We are interested in the average and maximum time taken for a population model to synchronise. By accumulating the reward along a path until some synchronised global state is reached we obtain a measure of the time taken to synchronise. Recall that we omit transitions between non-firing

[1] The scripts, along with the verification results, can be found at `https://github.com/PaulGainer/mc-bio-synch/tree/master/energy-analysis`

states; instead a transition is taken to the next global state where one or more oscillators do fire. For each transition from a firing state σ to some successor state σ', we define $\rho_{\mathsf{P}}(\sigma, \sigma') = \frac{1}{T}$. Given a non-firing state σ, let δ be the highest phase of any oscillator in that state. Hence, $T - \delta$ is the number of omitted transitions where no oscillators fire. Then, for each transition from σ to some successor state σ', we define $\rho_{\mathsf{P}}(\sigma, \sigma') = \frac{T-\delta}{T}$. In this way we obtain a measure of synchronisation time in cycles for a population model.

Power Consumption Let I_{id}, I_{rx}, and I_{tx} be the current draw in amperes for the idle, receive, and transmit modes, V be the voltage, C be the length of the oscillation cycle in seconds, and M_t be the time taken to transmit a synchronisation message in seconds. The power consumption in Watt-hours of one node for one discrete step within its refractory period, that is, the oscillator is in the idle mode, is $W_{id} = \frac{I_{id}VC}{3600T}$. Similarly, if the oscillator is outside of the refractory period, that is, it is in the receive mode, the corresponding power consumption is defined by $W_{rx} = \frac{I_{rx}VC}{3600T}$. Finally, let $W_{tx} = \frac{I_{tx}VM_t}{3600}$ be the power consumption in Watt-hours to transmit one synchronisation message. The power consumption of the network consists of the power necessary to transmit the synchronisation messages, and that of the oscillators in the idle and receive modes.

For synchronisation messages, we consider each firing state σ, and assign a reward of $\rho_{\mathsf{P}}(\sigma, \sigma') = k_1 W_{tx}$ to every transition from σ to a successor state $\sigma' = \langle k_1, \ldots, k_T \rangle$. This corresponds to the total power consumption for the transmission of k_1 synchronisation messages. For each firing state $\sigma = \langle k_1, \ldots, k_T \rangle$, the total power consumption for oscillators in the idle and receive modes is

$$\rho_Q(\sigma) = \sum_{\Phi=1}^{R} k_\Phi W_{id} + \sum_{\Phi=R+1}^{T} k_\Phi W_{rx} \ ,$$

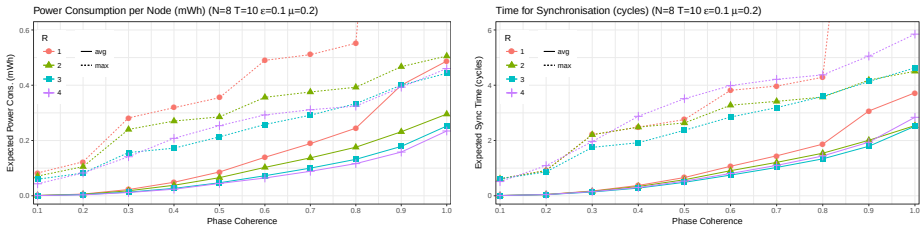
where R denotes the length of the refractory period. From a non-firing state σ the power consumed by the network to reach the next firing state is equivalent to the accumulation of the power consumption of the network in σ and any successive non-firing states that are omitted in the transition from σ to $\vec{\mathrm{succ}}(\sigma)$. For a non firing state $\sigma = \langle k_1, \ldots, k_T \rangle$ and maximal phase $\delta = \max\{\Phi \mid \Phi \in \{1, \ldots T\} \wedge k_\Phi > 0\}$ of any oscillator in that state, we define the reward as

$$\rho_Q(\sigma) = \sum_{j=0}^{(T-\delta)-1} \left(\sum_{\Phi=1}^{R-j} k_\Phi W_{id} + \sum_{\Phi=(R+1)-j}^{\delta} k_\Phi W_{rx} \right) .$$

The formula accumulates the power consumption over σ and subsequent $(T - \delta) - 1$ non-firing states, where the left and right summands accumulate the power consumption of nodes within, and outside of the refractory period, respectively.

5.2 Restabilisation

A network of oscillators is *restabilising* if it has reached a synchronised state, synchrony has been lost due to the occurrence of some external event, and



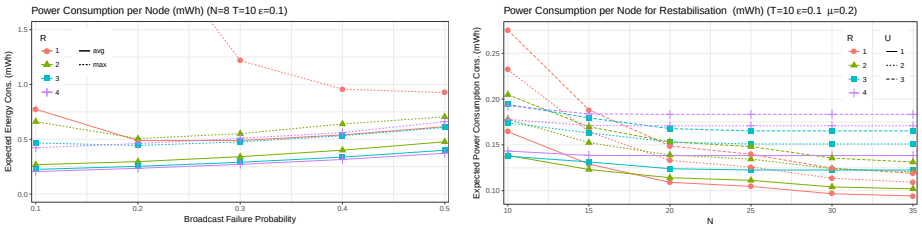
(a) (b)

Fig. 3. Power/Time per Node to Achieve Synchronisation

the network must then again achieve synchrony. We could, for instance, imagine the introduction of additional nodes with arbitrary phases to an established and synchronised network. While such a change is not explicitly encoded within our model, we can represent it by partitioning the set of oscillators into two subsets. We define the parameter U to be the number of oscillators with arbitrary phase values that have been introduced into a network of $N - U$ synchronised oscillators, or to be the number of oscillators in a network of N oscillators whose clocks have reset to an arbitrary value, where $U \in \mathbb{N}$ and $1 \leq U < N$. Destabilising U oscillators in this way results in configurations where *at least* $N-U$ oscillators are synchronised, since the destabilised oscillators may coincidentally be assigned the phase of the synchronised group. We can restrict the set of initial configurations by identifying the set $\Gamma_U = \{\langle k_1, \ldots, k_T\rangle \mid \langle k_1, \ldots, k_T\rangle \in \Gamma \text{ and } k_i \geq N - U \text{ for some } 1 \leq i \leq T\}$, where each $\sigma \in \Gamma_U$ is a configuration for the phases such that at least $N - U$ oscillators share some phase and the remaining oscillators have arbitrary phase values. As we decrease the value of U we also decrease the number of initial configurations for the phases of the oscillators. Since our model does not encode the loss or addition of oscillators we can observe that all global states where there are less than $N - U$ oscillators sharing the same phase are unreachable by any run of the system beginning in some state in Γ_U.

6 Evaluation

In this section, we present the model checking results for instantiations of the model given in the previous section. To that end, we instantiate the phase response function presented in Sect. 3 for a specific synchronisation model, and vary the length of the refractory period R, coupling constant ϵ, and the probability μ of broadcast failures. We use a synchronisation model where the perturbation induced by the firing of other oscillators is linear in the phase of the perturbed oscillator and the number of firing oscillators [15]. That is, $\Delta(\Phi, \alpha, \epsilon) = [\Phi{\cdot}\alpha{\cdot}\epsilon]$, where $[\_]$ denotes rounding of a value to the nearest integer. The coupling constant determines the slope of the linear dependency.



(a) (b)

Fig. 4. Power Consumption in Relation to Broadcast Failure Probability and Average Power Consumption for Resynchronisation to Network Size

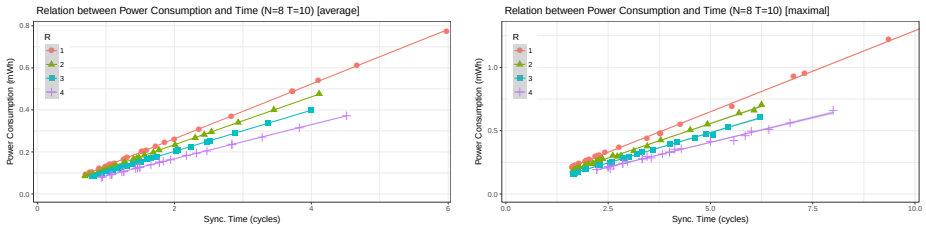
For many experiments we set $\epsilon = 0.1$ and $\mu = 0.2$. We could have conducted analyses for different values for these parameters. For a real system, the probability μ of broadcast failure occurrence is highly dependent on the deployment environment. For deployments in benign environments we would expect a relatively low rate of failure, for instance a WSN within city limits under controlled conditions, whilst a comparably high rate of failure would be expected in harsh environments such as a network of off-shore sensors below sea level. The coupling constant ϵ is a parameter of the system itself. Our results suggest that higher values for ϵ are always beneficial, however this is because we restrict our analysis to fully connected networks. High values for ϵ may be detrimental when considering different topologies, since firing nodes may perturb synchronised subcomponents of a network. However we defer such an analysis to future work.

As an example we analyse the power consumption for values taken from the datasheet of the *MICAz* mote [16]. For the transmit, receive and idling mode, we assume $I_{tx} = 17.4\ mA$, $I_{rx} = 19,7\ mA$, and $I_{id} = 20\mu A$, respectively. Furthermore, we assume that the oscillators use a voltage of 3.0 V. To analyse our models, we use the model checker PRISM [14] and specify the properties of interest in PCTL [10] extended with reward operators, allowing us to compute expected rewards for the reward structures defined in Sect. 5.1. For simplicity, we will omit the name of the reward structures when expressing properties.

Synchronisation of a whole network We analyse the power consumption and time for a fully connected network of eight oscillators with a cycle period of $T = 10$ to synchronise. Increasing the granularity of the cycle period, or the size of the network, beyond these values leads to models where it is infeasible to check properties due to time and memory constraints[2]. However, compared to our previous work [9], we were able to increase the network size.

Figures 3a and 3b show both the average and maximal power consumption per node (in mWh) and time (in cycles) needed to synchronise, in relation to the

[2] While most individual model checking runs finished within a minute, the cumulative model checking time over all analysed models was very large. The results shown in Fig. 3a already amount to 80 distinct runs.



(a) (b)

Fig. 5. Avg. (Max.) Power Consumption to Avg. (Max.) Time for Synchrony

phase coherence of the network with respect to different lengths of the refractory period, where $\epsilon = 0.1$ and $\mu = 0.2$. That is, they show how much power is consumed (time is needed, resp.) for a system in an arbitrary state to reach a state where some degree of phase coherence has been achieved. The corresponding PCTL properties are $\mathbf{R}^{\mathsf{avg}}_{=?}[\mathbf{F}\,\mathsf{coherent}_\lambda]$ and $\mathbf{R}^{\mathsf{max}}_{=?}[\mathbf{F}\,\mathsf{coherent}_\lambda]$, where $\mathsf{coherent}_\lambda$ is a predicate that holds for any state σ with $\mathsf{PCF}(\sigma) \geq \lambda$, and $\mathbf{R}^{\mathsf{avg}}$ and $\mathbf{R}^{\mathsf{max}}$ refer to the average and maximal reward accumulation respectively[3].

The much larger values obtained for $R = 1$ and phase coherence ≥ 0.9 are not shown here, to avoid distortion of the figures. The energy consumption for these values is roughly $2.4mWh$, while the time needed is around 19 cycles. Observe that we only show values for the refractory period R with $R < \frac{T}{2}$. For larger values of R not all runs synchronise [9], resulting in an infinitely large reward being accumulated for both the maximal and average cases. We do not provide results for the minimal power consumption (or time) as it is always zero, since we consider all initial configurations (global states) for oscillator phases. In particular, we consider the runs where the initial state is already synchronised.

As expected when starting from an arbitrary state, the time and power consumption increases monotonically with the order of synchrony to be achieved. On average, networks with longer refractory periods require less power for synchronisation, and take less time to achieve it. The only exception is that the average time to achieve synchrony with a refractory period of four is higher than for two and three. However, if lower phase coherence is sufficient then this trend is stable. In contrast, the maximal power consumption of networks with $R = 4$ is consistently higher than of networks with $R = 3$. In addition, the maximal time needed to achieve synchrony for networks with $R = 4$ is higher than for lower refractory periods, except when the phase coherence is greater than or equal to 0.9. We find that networks with a refractory period of three will need the smallest amount of time to synchronise, regardless of whether we consider the maximal or average values. Furthermore, the average power consumption for full synchronisation (phase coherence one) differs only slightly between $R = 3$ and $R = 4$ (less than 0.3 mWh). Hence, for the given example, $R = 3$ gives the

[3] Within PRISM this can be achieved by using the *filter* construct.

best results. These relationships are stable even for different broadcast failure probabilities μ, while the concrete values increase only slightly, as illustrated in Fig. 4a, which shows the power consumption for different μ when $\epsilon = 0.1$.

The general relationship between power consumption and time needed to synchronise is shown in Figs. 5a and 5b. Within these figures, we do not distinguish between different coupling constants and broadcast failure probabilities. We omit the two values for $R = 1$, $\epsilon = 0.1$ and $\mu \in \{0.1, 0.2\}$ in Fig. 5b to avoid distortion of the graph, since the low coupling strength and low probability of broadcast failure leads to longer synchronisation times and hence higher power consumption. While this might seem surprising it has been shown that uncertainty in discrete systems often aids convergence [9, 7].

The relationship between power consumption and time to synchronise is linear, and the slope of the relation decreases for higher refractory periods. While the linearity is almost perfect for the average values, the maximal values have larger variation. The figures again suggest that $R = 3$ is a sensible and reliable choice, since it provides the best stability of power consumption and time to synchronise. In particular, if the broadcast failure probability changes, the variations are less severe for $R = 3$ than for other refractory periods.

Resynchronisation of a small number of nodes We now analyse the power consumption if the number of redeployed nodes is small compared to the size of the network. The approach presented in Sect. 5.2 allows us to significantly increase the network size. In particular, the smallest network we analyse is already larger than that in the analysis above, while the largest is almost five times as large. This is possible because the model has a much smaller number of initial states.

The average power consumption per node for networks of size $10, 15, \ldots, 35$, where the oscillators are coupled with strength $\epsilon = 0.1$, and broadcast failure probability $\mu = 0.2$, is shown in Fig. 4b. The corresponding PCTL property is $\mathbf{R}^{\mathsf{avg}}_{=?}[\mathbf{F}\,\mathsf{coherent}_1]$, that is, we are only interested in the power consumption until the system is fully synchronised. The solid lines denote the results for a single redeployed node, while the dashed lines represent the results for the redeployment of two and three nodes, respectively. As expected, the more nodes need to resynchronise, the more energy is consumed. However, we can also extract that for higher refractory periods, the amount of energy needed is more or less stable, in particular, in case $R = 4$, which is already invariant for more than ten nodes. For smaller refractory periods, increasing the network size, decreases the average energy consumption. This behaviour can be explained as follows. The linear synchronisation model implies that oscillators with a higher phase value will be activated more and thus are more likely to fire. Hence, in general a larger network will force the node to resynchronise faster. The refractory period determines how large the network has to be for this effect to stabilise.

7 Conclusion

We presented a formal model to analyse power consumption in fully connected networks of PCOs. To that end, we extended an existing model for synchrony

convergence with a reward structure to reflect the energy consumption of wireless sensor nodes. Furthermore, we showed how to mitigate the state-space explosion typically encountered when model-checking. In particular, the model can be reduced by collapsing sequences of transitions where there are no interactions between oscillators. When investigating the restabilisation of a small number of oscillators in a network we can reduce the state space significantly, since only a small subset of the initial states needs to be considered. We used these techniques to analyse the power consumption for synchronisation and restabilisation of a network of MICAz motes, using the pulse-coupled oscillator model developed by Mirollo and Strogatz [17] with a linear phase response function. By using our model we were able to extend the size of the network compared with previous work [9] and discuss trade-offs between the time and power needed to synchronise for different lengths of the refractory period (or duty cycle).

Results obtained using these techniques can be used by designers of WSNs to estimate the overall energy efficiency of a network during its design phase. Unnecessary energy consumption can be identified and rectified before network deployment. Also, our results provide guidance for estimating the battery life of a network depending on the anticipated frequency of restabilisations. Of course, these considerations only hold for the maintenance task of synchronisation. The energy consumption of the functional behaviour has to be examined separately.

It is clear that our approach is inhibited by the usual limitation of exact probabilistic model checking for large-scale systems. We could overcome this by using approximated techniques, such as statistical model checking, or approaches based on fluid-flow approximation extended with rewards [4]. This would, of course, come at the expense of precision. An investigation of such a trade-off is deferred to future work. Our current approach is restricted to fully connected networks of oscillators. While this is sufficient to analyse the behaviour of strongly connected components within a network, further investigation is needed to assess different network topologies. To that end, we could use several interconnected population models thus modelling the interactions of the networks subcomponents. Furthermore, topologies that change over time are of particular interest. However, it is not obvious how we could extend our approach to consider such dynamic networks. The work of Lucarelli and Wang may serve as a starting point for further investigations [15]. Stochastic node failure, as well as more subtle models of energy consumption, present significant opportunities for future extensions. For example, in some cases, repeatedly powering nodes on and off over short periods of time might use considerably more power than leaving them on throughout.

References

1. Albers, S.: Energy-efficient algorithms. Commun. ACM 53(5), 86–96 (2010)
2. Bartocci, E., Corradini, F., Merelli, E., Tesei, L.: Detecting synchronisation of biological oscillators by model checking. Theor. Comput. Sci. 411(20), 1999–2018 (2010)
3. Bojic, I., Lipic, T., Kusek, M.: Scalability issues of firefly-based self-synchronization in collective adaptive systems. In: Proc. SASOW 2014. pp. 68–73. IEEE (2014)

4. Bortolussi, L., Hillston, J.: Efficient checking of individual rewards properties in markov population models. In: QAPL 2015. EPTCS, vol. 194, pp. 32–47. Open Publishing Association (2015)
5. Donaldson, A.F., Miller, A.: Symmetry reduction for probabilistic model checking using generic representatives. In: Proc. ATVA 2006. LNCS, vol. 4218, pp. 9–23. Springer (2006)
6. Emerson, E.A., Trefler, R.J.: From asymmetry to full symmetry: New techniques for symmetry reduction in model checking. In: Proc. CHARME 1999. LNCS, vol. 1703, pp. 142–156. Springer (1999)
7. Fatès, N.: Remarks on the cellular automaton global synchronisation problem. In: Proc. AUTOMATA 2015. LNCS, vol. 9099, pp. 113–126. Springer (2015)
8. Gainer, P., Dixon, C., Hustadt, U.: Probabilistic model checking of ant-based positionless swarming. In: Proc. TAROS 2016. LNCS, vol. 9716, pp. 127–138. Springer (2016)
9. Gainer, P., Linker, S., Dixon, C., Hustadt, U., Fisher, M.: Investigating parametric influence on discrete synchronisation protocols using quantitative model checking. In: Proc. QEST 2017. LNCS, vol. 10503, pp. 224–239. Springer (2017)
10. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. FAC 6(5), 512–535 (1994)
11. Heidarian, F., Schmaltz, J., Vaandrager, F.: Analysis of a clock synchronization protocol for wireless sensor networks. Theor. Comput. Sci. 413(1), 87–105 (2012)
12. Konishi, K., Kokame, H.: Synchronization of pulse-coupled oscillators with a refractory period and frequency distribution for a wireless sensor network. Chaos: An Interdisciplinary Journal of Nonlinear Science 18(3) (2008)
13. Kuramoto, Y.: Self-entrainment of a population of coupled non-linear oscillators. In: International Symposium on Mathematical Problems in Theoretical Physics. LNP, vol. 39, pp. 420–422. Springer (1975)
14. Kwiatkowska, M., Norman, G., Parker, D.: Prism 4.0: Verification of probabilistic real-time systems. In: Proc. CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer (2011)
15. Lucarelli, D., Wang, I.J., et al.: Decentralized synchronization protocols with nearest neighbor communication. In: Proc. SenSys 2004. pp. 62–68. ACM (2004)
16. MEMSIC Inc.: MICAz datasheet, `www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf`, last accessed 15th Jan, 2018
17. Mirollo, R.E., Strogatz, S.H.: Synchronization of pulse-coupled biological oscillators. SIAM J. App. Math. 50(6), 1645–1662 (1990)
18. Peskin, C.: Mathematical aspects of heart physiology. Courant Lecture Notes, Courant Institute of Mathematical Sciences, New York University (1975)
19. Rhee, I.K., Lee, J., Kim, J., Serpedin, E., Wu, Y.C.: Clock synchronization in wireless sensor networks: An overview. Sensors 9(1), 56–85 (2009)
20. Rhee, S., Seetharam, D., Liu, S.: Techniques for minimizing power consumption in low data-rate wireless sensor networks. In: Proc. WCNC 2004. pp. 1727–1731. IEEE (2004)
21. Soua, R., Minet, P.: A survey on energy efficient techniques in wireless sensor networks. In: Proc. WMNC 2011. pp. 1–9. IEEE (2011)
22. Wang, Y., Nuñez, F., Doyle, F.J.: Energy-efficient pulse-coupled synchronization strategy design for wireless sensor networks through reduced idle listening. IEEE Transactions on Signal Processing 60(10), 5293–5306 (2012)
23. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Computer Networks 52(12), 2292 – 2330 (2008)